

Evaluating the Generalisability of Segmentation Methods in Wifi-Sensing

Bachelor Thesis

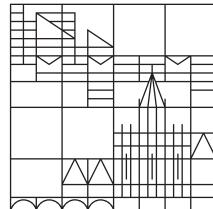
presented

by

Felix Dobler

at the

Universität
Konstanz



Faculty of Sciences
Department of Computer and Information Science

1. Evaluated by Prof. Dr. Daniel Keim
2. Evaluated by Prof. Dr. Heiko Hamann

Konstanz, 2024

Abstract

Wi-Fi has become more important in recent years besides the function of providing network connectivity. It is being repurposed for sensing through analysis of Channel State Information. Wi-Fi Sensing can be utilized in many sensing applications like presence and activity detection. One challenge of Wi-Fi Sensing is to overcome the fact that most approaches still require fixed hardware and are limited to their training environment. Segmentation of CSI is handled as one step towards efficient and accurate sensing. In this thesis, we test the interoperability of segmentation techniques on data collected with different hardware with a case study. An existing deep learning-based method "DeepSeg" is used to evaluate activity data in a new environment. With a maximum performance of 90% for segmentation, this approach promises compatibility for different data collection processes. Further, we highlight challenges in the robustness of existing methods and contribute our tools for public use.

Table of Contents

1. Introduction	1
2. Background	3
2.1. Wi-Fi	3
2.2. Channel State Information	3
2.3. Segmentation	5
2.3.1. Threshold based	5
2.3.2. Change Point Detection based	6
2.3.3. Deep Learning based	6
3. Related Work	8
3.1. DeepSeg: Deep-Learning-Based Activity Segmentation Framework for Activity Recognition Using WiFi	8
3.2. LightSeg: An Online and Low-Latency Activity Segmentation Method for Wi-Fi Sensing	11
4. Methodology	14
4.1. CSI Tool and Hardware	14
4.2. Experiment Setup	15
4.3. Preprocessing	16
4.4. Labeling	16
4.4.1. Labeling Tool	17
5. Adaptation and Replication of DeepSeg	18
5.1. Quality of Wi-Fi/CSI	18
5.1.1. CSI Frame Timings	18
5.1.2. Interference	19
5.1.2.1. Experiment: Reducing Interference	20
5.1.3. Automatic Gain Control	20
5.2. Modifications to DeepSeg	22
5.2.1. DeepSeg Improvements	22
5.2.2. Data Transformation	22
5.3. Extend Collected Data	23

6. Results	24
6.1. Sensing Performance	24
6.1.1. Mixing of Antenna Pairs	24
6.2. DeepSeg Performance Robustness	24
6.2.1. Test Data Dependence	25
6.2.2. Reduced DeepSeg Training Data	25
6.2.3. Hyperparameter optimization	25
7. Discussion	27
7.1. Training Data Sensitivity	27
7.2. Generalisability of Segmentation Methods	28
7.3. Limitations	29
7.3.1. Insufficient User Variety	29
7.3.2. Data Quality	29
7.3.3. CSI Antenna and Subcarrier Selection	29
7.3.4. Joint Training of Segmentation and Classification	30
7.3.5. Wi-Fi Interference and Frame Timings	30
7.4. Future Work	30
7.5. Conclusion	31
Bibliography	32
A. Training Parameters and Performance	37

CHAPTER 1

Introduction

Wi-Fi Signals are nowadays basically everywhere. On the way from the Wi-Fi transmitter, the signals are influenced by all kinds of objects. They can be reflected, absorbed, and scattered by the environment (Figure 1). With Wi-Fi Sensing, we can capture and interpret relevant information about the surroundings by leveraging the propagation characteristics of Wi-Fi Signals.

Because the interactions of Wi-Fi signals highly depend on the surroundings, machine learning models for their specific applications are often tailored to their training environment. To apply it in a different location or even a changed environment, the training of machine learning models has to be conducted again for the specific environment. To perform the computationally expensive model training would require collection of a custom dataset and generate high service/energy costs [9]. As the result is a model suitable for only one primary environment, this approach is not exactly sustainable.

Wi-Fi Sensing is of high importance due to its promising use in many different areas [29]. Presence detection [28], person localization [13], fall detection [13], activity and gesture recognition [13, 15, 17, 23, 24, 32, 36, 39, 52], and more, have been successfully demonstrated.

The data source for Wi-Fi Sensing is nowadays mainly Channel State Information (CSI), which can be captured using certain consumer-grade network interface cards. It is therefore very cost-effective, relative to techniques like mmWave-based sensing, which is a sensing technique based on much higher frequency wireless signals. The use of Wi-Fi signals also means that data can be captured in challenging environments for cameras, such as in low light and even beyond visual line-of-sight [39].

To leverage all these benefits in a realistic setting that goes beyond lab environments, an efficient way to detect timespans of activity is needed. This process of detecting the bounds of activities is called segmentation. CSI data for research is often collected by starting the recording, performing an activity, and then stopping the recording [31, 37, 50]. This grants a data sequence that only contains the target activity and no or not much unnecessary data. Applications like fall detection, intrusion detection, gesture controls, and more will likely be active continuously. Segmentation has a large potential to contribute to the real-world feasibility of Wi-Fi Sensing solutions by enabling accurate classification [46, 9]. Computational cost and requirements can also be improved through the use of low-power algorithms [9].

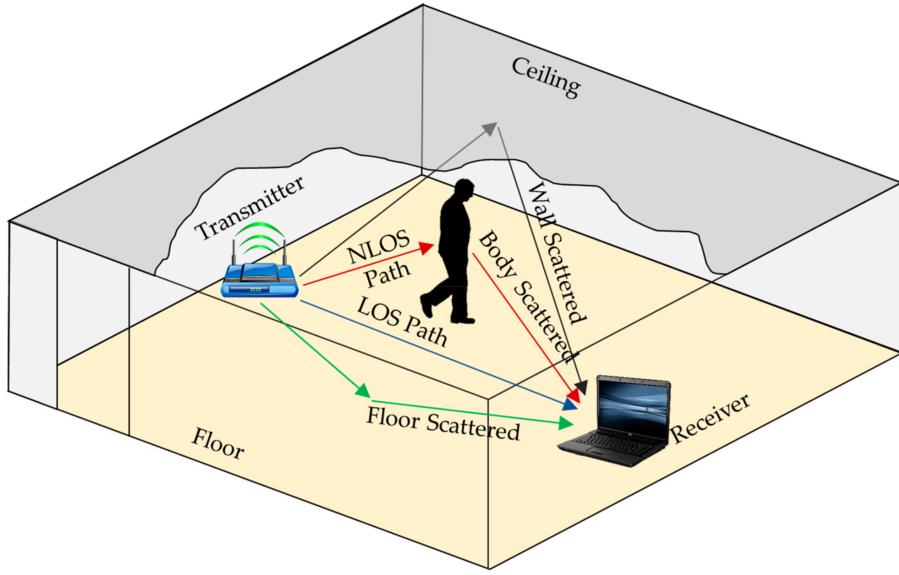


Figure 1. Working Principle of Wi-Fi Sensing

Reproduced from [19] with permission, licensed under CC BY 4.0

Such approaches are designed to be lightweight and fast, only triggering the more extensive activity classification once the activity segment bounds are found.

CSI is very dependent on the specific environment and approaches are often implicitly developed to only operate on datasets that were collected with the same hardware as the training data.

The goal of this thesis is to improve on the aspects of environment independence and hardware inter-operability, leading to generalizability of segmentation methods. The current progress in this area of state-of-the-art segmentation models, is examined by conducting a case study on DeepSeg.

For this purpose, we first collect a dataset of activities (4), investigate the wifi environment and apply required modifications to DeepSeg (5). Next, a neural network is trained on the dataset and evaluated (6). The results are discussed and finally put into the scope of the research question and future research 7.

CHAPTER 2

Background

Wi-Fi Sensing leverages the influence of changes, like activity of a human, in an environment on the propagation characteristics of Wi-Fi signals. With these recorded characteristics, machine learning can be applied to extract data for various applications. To detect the changes in signals, Wi-Fi Sensing setups consist of a transmitter that emits Wi-Fi signals and at least one receiver that measures signal characteristics of received frames.

2.1 Wi-Fi

The Wi-Fi specification divides the frequencies used for Wi-Fi communication into channels.

These channels cover the frequency spectrum in the 2.4 GHz or, respectively, the 5 GHz band. A channel's width can reach from 20 and 40 MHz in the 2.4 GHz band up to 160 MHz in the 5 GHz band. The use of orthogonal frequency-division multiplexing (OFDM) divides a channel further into subcarriers. A 20 and 40 MHz channel width results in 56 and 112 data and pilot subcarriers, respectively.

Multiple-input and multiple-output (MIMO) is a principle used in many wireless communications, enabling communication between devices using multiple antennas. The number of antennas is specified by " $M \times N$ MIMO" where M is the number of transmitting and N of receiving antennas.

2.2 Channel State Information

CSI describes properties of a wireless link. It contains information about how the signals travel from a transmitter to a receiver and the external influences they experience on the way. With the help of CSI, the transmitter can modify its parameters to improve the link quality, thereby enhancing latency, transmission speeds, and reliability.

In the beginning of Wi-Fi Sensing, the Received Signal Strength Indication (RSSI) was used [19, 51]. RSSI is a straightforward metric that indicates the overall strength of the signal. It is insufficient for inferring meaningful information regarding multi-

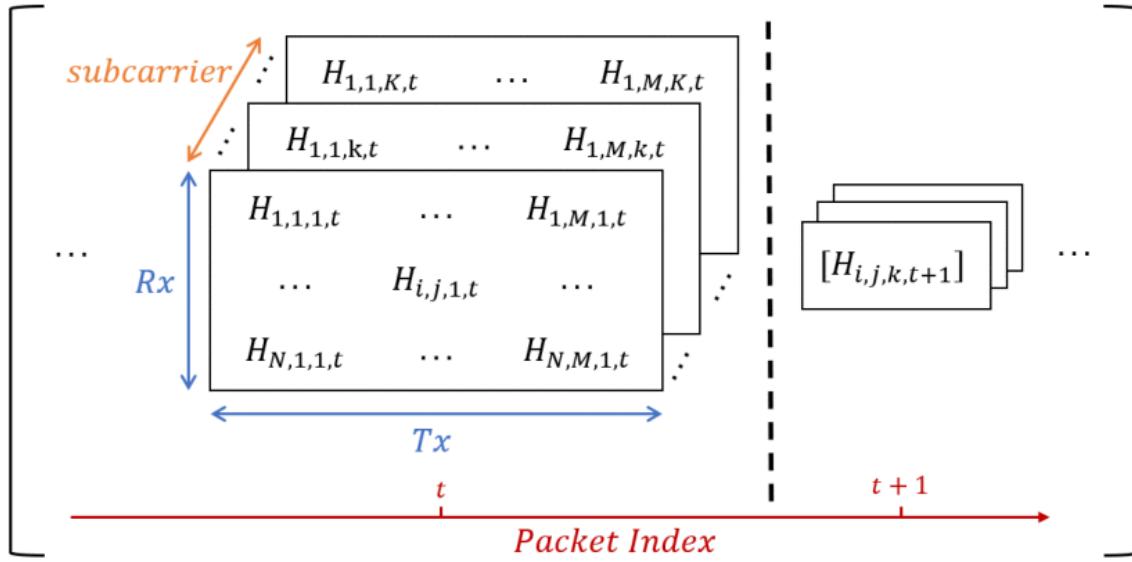


Figure 2. Structure of CSI data,

Reproduced from [19] with permission, licensed under CC BY 4.0

path environments, where the signals reach the receiver on multiple paths by interacting with the environment [13]. In contrast to this, CSI provides much more detailed data (Figure 2).

CSI can be calculated for individual Wi-Fi packets. Performing this calculation on consecutive packets results in a time series of CSI. The CSI of a single packet is represented by a complex 3D matrix H with the dimensions Rx , Tx , and subcarriers. Rx is introduced by the number of receiving N antennas. The M transmitting antennas create the dimension Tx . The subcarrier dimension contains the K different subcarriers. This number depends on the network's channel width and the CSI report grouping (subsection 5.2.2) [22].

The channel state of a Wi-Fi connection is commonly modeled as $y = Hx + n$. y is the measured signal at the receiver, H the CSI matrix, x the sent signal from the transmitter and n the noise vector. The value of x is defined in the Wi-Fi specification [22] and therefore known by both parties in advance. Because the receiver knows that the transmitter sent signal x , but the signal y was measured at the receiver, it can estimate the CSI [30]. Each element in the matrix is a complex number for which, excluding zero values, the amplitude and phase can be calculated. The structure is visualized in Figure 2.

The part of Wi-Fi frames used for channel estimation is the "long training field" inside the preamble [22]. Taking the role of the known symbol x from the equation above, it enables the estimation of the CSI matrix.

The amplitude component of CSI is more suitable for direct use, whereas the phase information of the measured signal requires extensive preprocessing. Advanced information, such as the "Angle of Attack" and differences in the "Time of flight" (signal travel duration), can be extracted from phase information. As summarized in [43, 30], mismatches in the transmitter's and receiver's clocks and frequencies introduce

an accumulating offset in the phase, which requires correction. Therefore, using the additional information of the phase property requires increased preprocessing and is less often used.

CSI is usually not exposed to the user in an easy-to-use format. Additional steps are needed to obtain this data. They consist of:

1. Using a device for wireless connectivity that exposes the CSI to the system
2. With a compatible tool, acquire the CSI from the device

The tools usually only work with a single product family/manufacturer [20, 35, 47].

2.3 Segmentation

The need to split a time series into multiple discrete segments is not exclusive to Wi-Fi Sensing. It is present in several other fields, including medical monitoring, speech recognition, image analysis, and general non-Wi-Fi based human activity recognition [5]. The task of segmenting CSI is particularly new and does not incorporate all advances in the segmentation field contributed by other research areas.

For Wi-Fi Sensing, three major activity segmentation techniques have been used so far to our knowledge. They provide individual benefits and drawbacks and will be introduced in the following.

2.3.1 Threshold based

Performing an activity in the range of Wi-Fi devices usually results in a higher variance in CSI amplitudes than in a static environment. This change can be used to detect the start and end points of activities. A straightforward approach is to use a threshold. If the variance of the CSI crosses a certain predefined value, the underlying CSI is rated as part of an activity.

While there are modified versions of this threshold concept, such as those used for performance comparison in section 3.1, they mostly divide into fixed- and dynamic threshold categories [46, 9].

Fixed threshold segmentation is based on (often manually) determined thresholds [40]. The threshold value has to be chosen based on the usual noise levels of the environment. This makes human analysis of the CSI necessary, and the selected threshold is not steady in its performance when environmental changes are introduced. Reduced segmentation accuracy or the need for threshold readjustment arises as a consequence [9, 7, 38].

On the other hand, activity segmentation with a dynamic threshold uses a threshold that can adapt to the data. Approaches include, e.g., setting the dynamic

threshold to a fixed multiple of the current noise level [41]. While better at dealing with foreign environments without recalibration, this approach still cannot adjust to changes in the ratio between noise and activity amplitudes, as they exist with different activity coarseness levels.

2.3.2 Change Point Detection based

This method focuses on detecting points in time at which a change of properties or characteristics occurs, not just the raw CSI amplitude or variance. Although Change Point Detection (CPD) can achieve good results in various applications, they are usually still limited by a problem similar to static-threshold segmentation. They internally still use a threshold to determine whether the change score of a sample is large enough to be considered a change point in the data. Because of that, they also face challenges when applied to domain-specific or considerably changing data [4].

2.3.3 Deep Learning based

Both of the above approaches also struggle with another condition, mixed-granularity activities. Not all general activities induce similar changes in the magnitude of CSI data. For example, a person's fall will result in a much bigger perturbation than raising an arm.

Threshold-based segmentation suffers a considerable performance penalty in this task because a single threshold can not sufficiently be used to work on different granularities, which might even have overlapping "magnitudes" counting as part of an activity on some occasions but not others [46, 9].

Coarse-grained activities have more variance in the CSI data, possibly including sections of medium variance leading up to and trailing the core activity. This requires a high threshold for correct segmentation.

Fine-grained activities can have a small-variance lead-up and tail-off, with a medium-variance activity. Here, a smaller threshold is required to segment the activity. Choosing one threshold value to segment both activity granularities is insufficient as it would either include too much of the coarse-grained activity or preserve too little of the fine-grained one. This problem is visualized in Figure 3. Striving to perform activity classification on these mixed-type activities brings up the need for more flexible activity segmentation techniques.

Applying deep learning to the activity segmentation problem provides the potential to learn the patterns in CSI data that indicate an activity segment. Using convolutional neural networks, the decision process can be made considering the behavior of the CSI with more surrounding context while being flexible enough to deal with noise and changing environments.

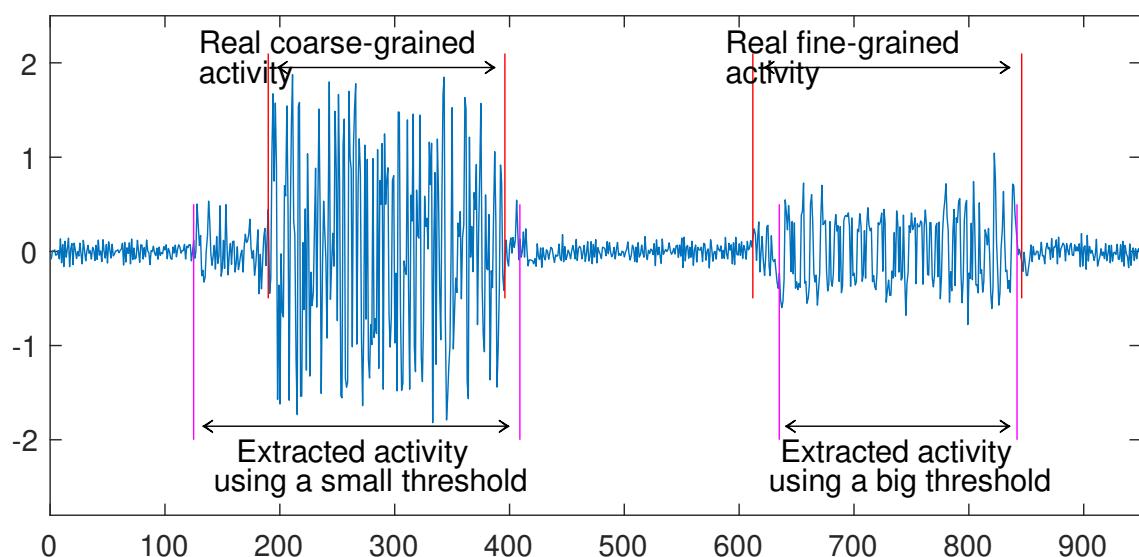


Figure 3. Challenge of Threshold-based Mixed-activity Segmentation,
Reproduced from [46] with permission, © 2021 IEEE

CHAPTER 3

Related Work

Wi-Fi Sensing presents itself as a potential solution to various applications.

Among more common sensing tasks, Wi-Fi Sensing has also been used for tasks of progressive complexity like sign language recognition [21, 31, 37, 45, 49], person identification [25], keystroke detection [27, 2, 3], and respiratory monitoring [42].

As the core methods have matured, the research focus now shifts to the surrounding challenges. The strong influence of different environments on the measured data is a crucial hindrance in reaching the general usability of developed models. This includes the characteristic influence of the physical surroundings affecting the propagation of Wi-Fi signals and introducing noise, as well as the limited support of Wi-Fi devices capable of providing CSI. With [39, 25, 32], recent advances have been made concerning the robustness of Wi-Fi Sensing when presented with new environments. By using training data from multiple recording environments, altering device positions between recordings, and implementing few-shot learning to adapt a model to the target environment, the stability for new environments improved.

Another sub-area of Wi-Fi Sensing with limited research is the segmentation of CSI data. For the goal of substantial real-world adoption, reliance on human complementary work in the sensing process should be minimized. Because of the comparatively early state of Wi-Fi Sensing, many research projects use manually segmented CSI. Only through automation of this step can approaches that require it efficiently and quickly perform their task (e.g., fall detection).

Our work focuses on a combination of these two core challenges: The viability of automated segmentation when applied in a different physical environment and with data collected through means of different hardware. The following two papers represent the current state-of-the-art segmentation techniques for activity recognition in Wi-Fi Sensing.

3.1 DeepSeg: Deep-Learning-Based Activity Segmentation

Framework for Activity Recognition Using WiFi

Xiao et al. try to solve the challenges of traditional methods, previously mentioned in 2.3.3, by modeling the detection of activity bounds as a classification problem and applying Deep Learning (DL) methods. Furthermore, they present an accompanying activity classifier and use a feedback mechanism to the segmentation network,

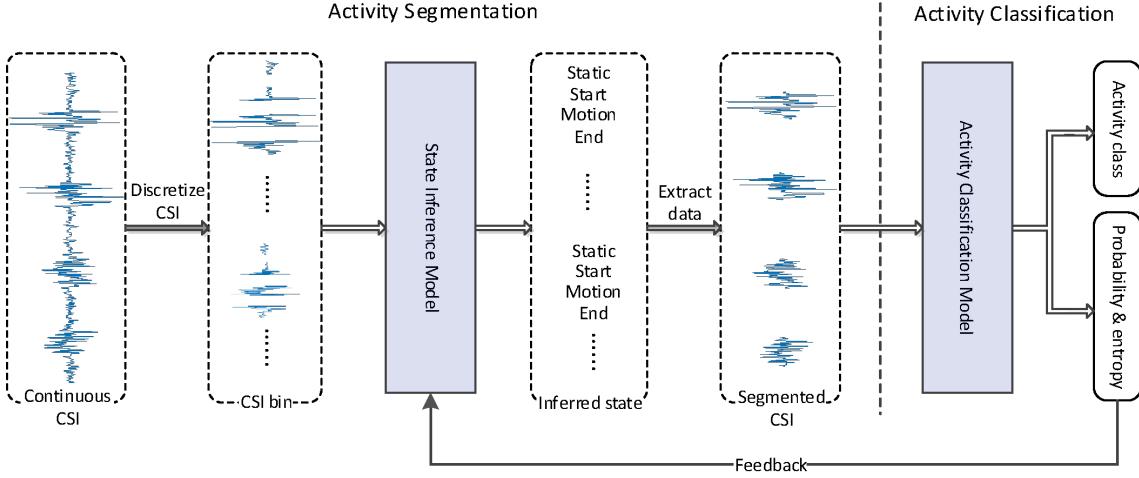


Figure 4. Structure of the DeepSeg architecture
Reproduced from [46] with permission, © 2021 IEEE

further improving its performance. The performed experiment demonstrates notable improvements over state-of-the-art approaches.

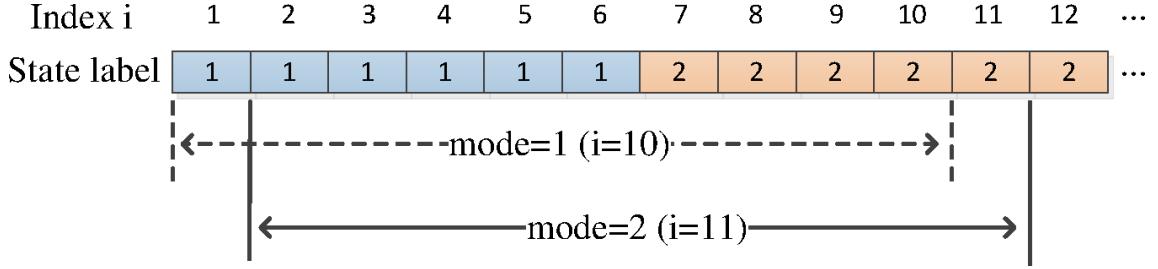
DeepSeg is a two-stage approach. The first stage identifies the activity segment bounds, while the second classifies the performed activity. The overall structure is depicted in Figure 4.

Detecting the activity segments is done by training a Convolutional Neural Network (CNN) classifier and applying that to data generated from a sliding window over the CSI stream. The classes for the CSI are: static-state, start-state, motion-state, and end-state. They indicate idle phases, the start of an activity, the ongoing activity, and the end of an activity, respectively. The training data for this model is generated by extracting specific sections of CSI from recordings, using manually segmented data as ground truth. For every performed activity, the segment start and end labels serve as middle points for the start-state and end-state. The motion state is extracted from the middle of the activity, and the static state is extracted directly after the end state.

The size of these sections is determined by the configurable window size w .

After training the model, it can then be used to detect the segments in CSI streams. This is done by taking the stream and extracting bins of size w using a sliding step size of 1. Next, the state inference model predicts each bin's state. We then use another sliding window of m predicted labels and apply the mode operation to them. If the mode of the window changes, it marks a transition point in the underlying data. Figure 5 shows an example of detecting a segment start point.

A changing mode from 1 to 2 indicates a transition from *static-state* to *start-state* and therefore, index 7 in the middle of the mode-sliding-window is chosen as the start of an activity segment.

**Figure 5.** Example of starting point detection

Reproduced from [46] with permission, © 2021 IEEE

The activity classification model uses the same basic architecture as the state inference model, consisting of multiple convolutional, dropout, fully connected, and pooling layers.

DeepSeg uses a feedback mechanism from the classifier to the state inference model to further improve the latter one's performance. After a phase where both models are pre-trained individually, they are jointly trained with the classifier, which provides a concentration degree that incorporates how confident the model is in its activity prediction. This value is then used as a modifier for the segmentation stage. Samples that resulted in a high score are given a higher weight in training. This trains the segmentation model to generate results that better represent an activity.

Data collection was set up as follows. They used a commercial router and a Laptop equipped with an Intel 5300 NIC (3 Antennas). Combined with the Linux 802.11n CSI Tool [20], they acquired CSI measurements with 1000 samples/s, which were down-sampled to 50 per second after applying a low-pass filter. Five volunteers each performed a set of 10 activities, of which five were coarse-grained (boxing, picking up, running, squatting, walking) and five were fine-grained (hand swing, hand raising, pushing, drawing O, drawing X). In each recording, one user performs a sequence of a fine-grained and then a coarse-grained activity, each five times with small pauses between all activities.

Every user performs each activity 30 times, resulting in 1500 activity samples in total. The train/test ratio is reported to be 4 : 1.

The resulting performance is compared against three threshold-based segmentation models [40, 44, 18] and three CPD-based models [8, 6, 5]. Further, the influence of training the model with

- DeepSeg feedback mechanism used or without
- only fine-, only coarse-grained or mixed activity
- different window sizes w
- different training dataset sizes

was explored.

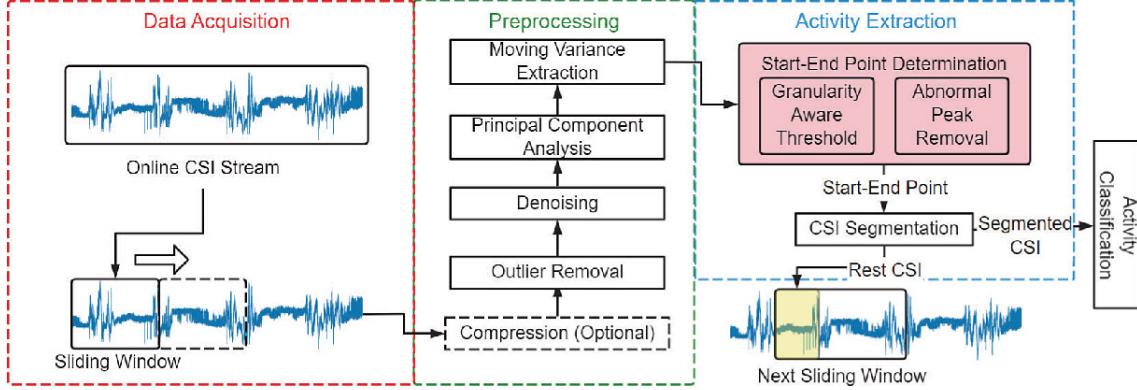


Figure 6. Structure of LightSeg
Reproduced from [9] with permission from Springer Nature

Using a custom accuracy metric $|A \cap B|/\max\{|A|, |B|\}$ with A representing the activity's manually determined ground truth start and end points, and B the predicted segment bounds.

For mixed-granularity activities, DeepSeg's segmentation model achieved an accuracy of about 92%, outperforming every compared segmentation approach. The accuracy of the activity classification model reached about 94%.

3.2 LightSeg: An Online and Low-Latency Activity Segmentation Method for Wi-Fi Sensing

In 2023, Chen et al. published a new segmentation method called "LightSeg" [9]. This mixed-granularity compatible segmentation technique claims to be usable online and with low latency, meaning it's intended to be applied live on the CSI, not in post-recording analysis.

The paper also addresses the challenges of previous Wi-Fi Sensing segmentation algorithms, stating that threshold-based methods require predefined thresholds that are not applicable across varying environments and the shortcomings with mixed-granularity activities. Additionally, they identified a problem area of deep learning approaches: the need for significant training datasets and computing power. This makes DL unsuitable for low-latency and on-client segmentation, demanding high-performance computing infrastructure instead.

LightSeg uses a combination of multiple common preprocessing steps on the utilized CSI amplitude data: outlier removal, discrete wavelet transform for denoising, dimensionality reduction with PCA, and moving variance extraction.(The architecture of LightSeg is shown in Figure 6.) As an option for resource-limited hardware, equidistant sampling is supported to reduce the computational requirements with the trade-off of less accurate segmentation results. The effects of preprocessing can be traced in Figure 7.

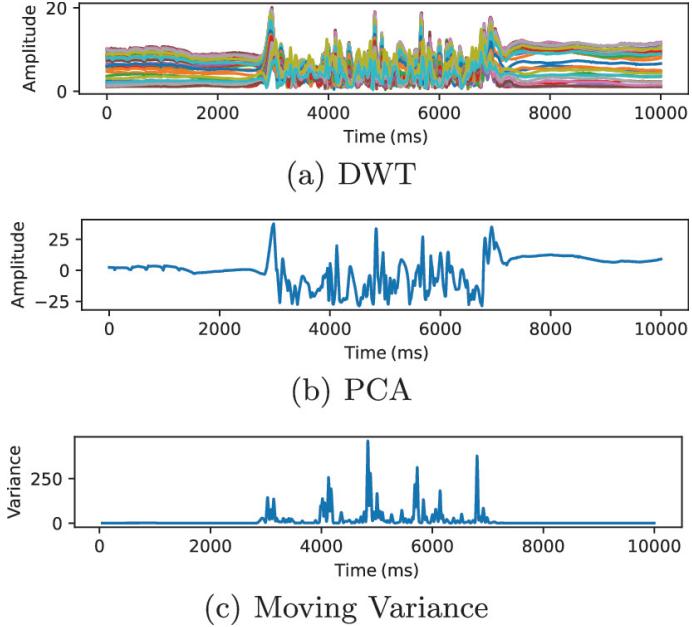


Figure 7. Effects of Preprocessing in LightSeg
Reproduced from [9] with permission from Springer Nature

The introduced segmentation algorithm consumes the live CSI variance data from a buffer with a sliding window. Within this window, it tries to detect the endpoint of an activity first, utilizing a fraction of the window’s maximum variance as a threshold. If an endpoint is found, it works backward to locate the beginning of the activity using a suitable threshold of the already traversed CSI. Chen et al. encountered abnormal peaks in the data, which they attributed to environment changes, automatic gain control, or even transceiver errors. Due to the short nature of the noise, they added abnormal peak removal into the process. This activity-dependent threshold enables LightSeg to segment data with mixed-granularity activities more accurately than simple threshold methods.

The approach was evaluated using DeepSeg’s published dataset and segmentation accuracy metric. LightSeg outperforms DeepSeg in mixed-, coarse-, and fine-granularity activities by about 1% (scoring 93% for mixed-type sequences). Comparison against Wi-Multi [18] falls even more in LightSeg’s favor, with improvements in accuracy between 4% to 6.4% for the different activity-granularities. Activity classification accuracy is very similar to DeepSeg. Only very minor (sub 1%) advances could be measured.

Besides achieving higher segmentation accuracy, LightSeg features drastically higher computational efficiency for segmenting activities. Chen et al. also demonstrated the environment-independent thresholding approach by collecting a new dataset containing mixed-granularity activities and demonstrating similar segmentation performance on both datasets. In a runtime-based comparison of the two approaches (using only CPU), LightSeg only took 0.3 s to segment an activity, in contrast to 10 s for DeepSeg. LightSeg requires about 3% of the computing time

and is more memory efficient than DeepSeg, presenting itself as a viable alternative for mixed-granularity segmentation.

CHAPTER 4

Methodology

For our experiment, we selected DeepSeg as a foundation for our work. While LightSeg claims to offer slightly better performance, its implementation details are unavailable, and it relies on DeepSeg's classification CNN. As a result, both projects would need to be reproduced for evaluation. For us, DeepSeg provides a more accessible starting point, because of the published code and datasets. This chapter outlines our decisions for data collection, data processing, and the modifications applied to DeepSeg.

4.1 CSI Tool and Hardware

As a system for obtaining CSI data, we selected the "Atheros CSI Tool" [33, 47] in combination with two "NETELY NET-N450A" Wi-Fi network interface cards. They are based on the Qualcomm Atheros AR9590 chip, which is supported by the CSI tool.

Following the goal of testing the generalisability of Wi-Fi Sensing, we chose a different tool than Xie et al. who used the "Linux CSI Tool" [20]. This will allow us to determine whether a segmentation approach designed for specific hardware and software is suitable for use with different data collection methods.

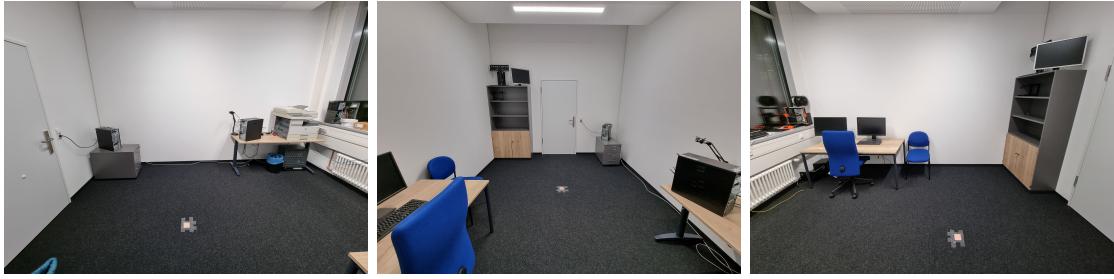
We specifically selected the Atheros CSI Tool because it provides a larger amount of data. The key capabilities of both tools are compared in Table 1. One of the benefits was the availability of compatible Wi-Fi-Chipsets. The Intel 5300 NIC was only available from local vendors in "used" condition, which could've had an unpredictable effect on the captured data. Another advantage over the Linux CSI

Table 1. Comparison between Intel and Atheros CSI Tool [20, 47]

CSI Tool	Linux 801.11n CSI Tool	Atheros CSI Tool
Device	Intel 5300 NIC	Qualcomm Atheros Chipsets
Subcarriers	30	up to 114
Bit Depth	8 Bit	10 Bit
MIMO Configuration	3x1	3x3
Firmware	Custom Firmware	Stock Firmware
Compensates AGC	Yes	No

Table 2. Recorded Activity Sequences

Sequence	Activity Type 1	Activity Type 2
iw	Boxing	Hand Swing
ph	Picking Up	Hand Raising
rp	Running	Pushing
sd	Squatting	Drawing O
wd	Walking	Drawing X



(a) User's Orientation

(b) Right View

(c) Back View

Figure 8. Photos of Recording Room

Tool is the higher amount of available data. Instead of returning CSI for only 30 subcarriers at 8-bit resolution with a maximum of 3x1 MIMO, Atheros CSI Tool provides up to 114 subcarriers, 10-bit resolution, and 3x3 MIMO. Additionally, the tool for Atheros chipsets doesn't require modification of the Wi-Fi-NIC's firmware.

4.2 Experiment Setup

Activity data was collected in the form of CSI and video recordings. Due to time constraints, we could only record a single user. They performed the identical types of sequences as described in DeepSeg, each consisting of five coarse- and five fine-grained activities consecutively. The different types of sequences are outlined in Table 2.

This experiment was conducted in an office room, photos of which are shown in Figure 8. This office is situated in a row of other rooms, all connected by a hallway. Minor influences on our data through activities outside the recording room are possible.

The setup consists of two desktop PCs positioned alongside a wall. Following DeepSeg's guidelines, the PCs were configured with Ubuntu 14 and configured to use the compiled Linux kernel with the modified ath9k driver. The transmitting and receiving PCs are on the user's front left and front right, respectively. A layout of the room can be seen in Figure 9. The devices are oriented such that the Wi-Fi card's antennas are aligned with the user.

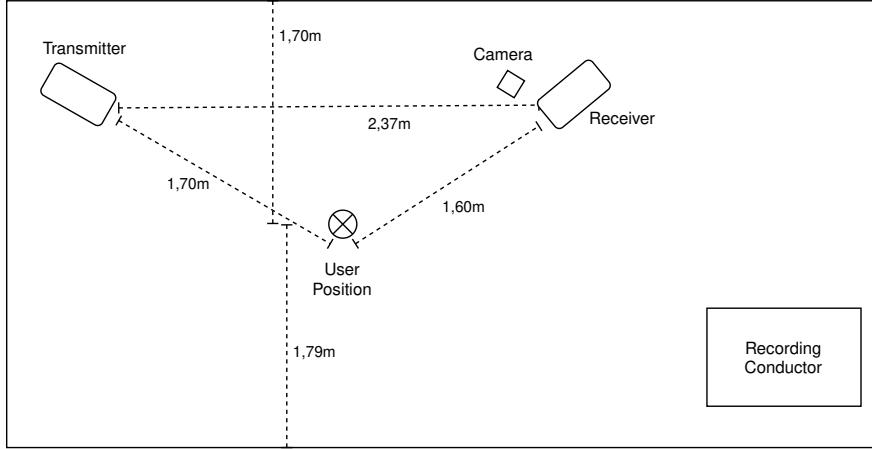


Figure 9. Recording Room Layout
Not to Scale

Additionally, videos of the user performing the activities were recorded to provide accurate ground truth when labeling the CSI data. The camera was connected to the receiving device and positioned directly beside it.

During each recording, the user performed the given activity sequence on top of the marked position on the ground. The recording conductor was also present in the room to monitor the execution of the tests and start/stop the recording. They moved as little as possible during the recordings to minimize their distortions in the data.

Our data set consists of 5 sequence types with 6 recordings each, containing 10 activities per sequence (5×2 types). After excluding one faulty sequence, this results in 290 activity samples.

4.3 Preprocessing

Following DeepSeg, a low-pass Butterworth filter is applied to the data along the time axis to reduce high-frequency noise. The filter has a cutoff frequency of 25 Hz. Next, the data is down-sampled to retain every 20th sample. This reduces the amount of data involved in the later computationally expensive stages, including CNN training and inferencing.

4.4 Labeling

With the help of the simultaneously recorded video data, the CSI sequences were manually labeled. This consisted of marking the individual activities' start and end

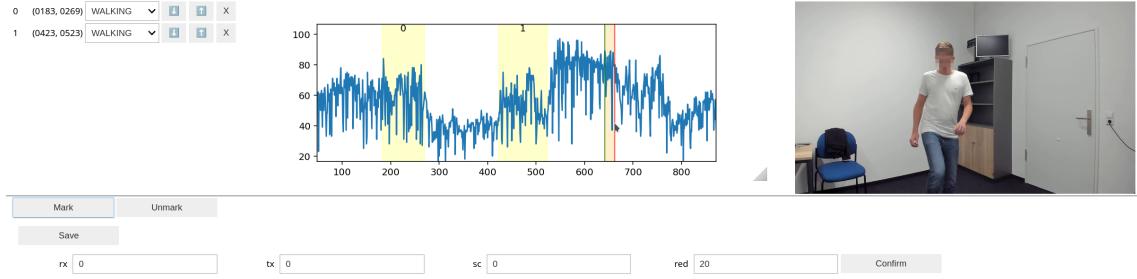


Figure 10. Tool for Labeling of CSI Based on Video Recordings

points and the activity type. The labeling tool described in subsection 4.4.1 was used.

Because the CSI data is down-sampled by factor 20 for use with the CNN after the first preprocessing step, the target CSI rate is now 50 samples/s. The labeled activities have a mean length of 100.1 frames with a standard deviation of 14.1 frames.

4.4.1 Labeling Tool

To label the collected data, a Python-based tool was developed. It enables us to load a CSI sequence, select the antenna pair and subcarrier, and then display the CSI amplitude for the chosen values. The sequence type (including the two performed activity types) is automatically detected from the filename of the CSI recording. A cursor can be moved through the CSI data. Mouse clicks on the CSI graph mark the current cursor position as an activity's start or end point. Next to the CSI visualization, the video frame matching the timestamp of the cursor's position in the plot will be displayed. This allows us to confidently identify the activity segments in the video and label the corresponding CSI data.

Marked segments will be displayed in the plot and in a list where they can be removed, reordered, and assigned the correct activity type. The labeling tool is shown in Figure 10.

CHAPTER 5

Adaptation and Replication of DeepSeg

This chapter covers the changes made to DeepSeg to reach compatibility with data from the Atheros CSI Tool. Further, it documents the encountered challenges regarding the Wi-Fi setup and environment. Our dataset, code containing adjustments, and further information will be available for public use [16].

5.1 Quality of Wi-Fi/CSI

This documents the state and configuration of the Wi-Fi devices to aid with better reproducibility and potential improvements of the experiment.

The choice of Wi-Fi hardware and corresponding CSI Tool is already outlined in section 4.1. To conduct measurements using the Atheros CSI Tool, the device transmitting the Wi-Fi frames first creates a wireless network. The receiver then connects to this network and can then receive frames sent by the transmitter and calculate the CSI for them.

5.1.1 CSI Frame Timings

The sending frequency of the transmitting device was lowered from the default (limited by connection throughput) to 1 kHz. Delays in both the sender's and receiver's processing chain as well as Wi-Fi latency reduced this frequency to an average of 910 samples/s with a standard deviation of 6.2 frames/s across the different measurements.

Further, irregular intervals between successive frames were noticed. Occasionally, there would be spikes in the frame time, such as one frame taking a long time to be sent to the client, but then the following one would arrive very quickly. The logged intervals on the sender's side were mostly regular. Both devices' intervals are visualized in Figure 11)

Frame timings can only be monitored in the user-space code of the Atheros CSI Tool on both the sender and receiver. For a more accurate analysis, it would be necessary to use the timings of frames from within the modified ath9k driver, which runs in the kernel. Those timings are not accessible by default. According to the

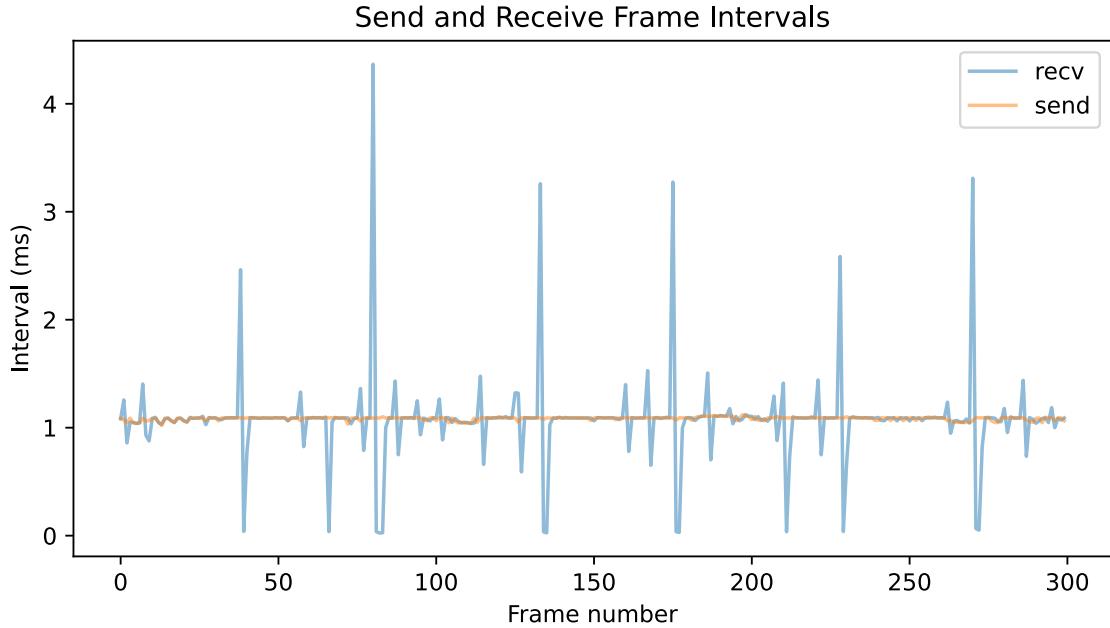


Figure 11. Irregular CSI Intervals at the Receiver

documentation, the timestamp when a frame was received should be part of the data returned. The data provided by the software had a blank timestamp field. The field value was set to the current timestamp when the frame was processed in the receiver's user-space code

A plausible cause of the occasional spikes in frame times would be that while the sender's user space program passes the frames to the Wi-Fi card's driver in a regular matter, the driver does not send all data immediately. When a Wi-Fi client detects activity on the channel that it wants to send on, it will hold off for a short time, to then retry to send at a later time [22]. This mechanism is called "CSMA/CA". Waiting to transmit frame number n can result in the next frame number $n + 1$ being submitted to the Wi-Fi device's buffer before it can send frame n . As a result, frame n on the receiver would have a high frame time, followed by a low frame time for $n + 1$, as both frames could be sent together in one transmission.

5.1.2 Interference

The Wi-Fi network used for the recordings will only be affected by other networks using the same or neighboring channels. The PC broadcasting the network was set to use channel six, so other networks on channel six will cause the strongest interference. Because the channel width (20 MHz is higher than the spacing between channels (5 MHz) [22], even networks on channels [2; 10] can cause interference, albeit not as significant in effect. An environment with less ambient Wi-Fi activity should result in more reliable frame timings.

Because of the busy Wi-Fi environment, the tool in use ("hostapd") for creating the access point refuses to create a network with 40 MHz channel width. This is done in adherence to the Wi-Fi specification, which commands this behavior under the section on Overlapping Basic Service Sets (OBSS) [22]. The reduced channel width results in CSI for only 56 instead of 114 subcarriers. A higher channel width would likely result in more valuable data, as the subcarrier frequencies could cover a higher range and convey more information about the environment in the CSI.

5.1.2.1 Experiment: Reducing Interference

In an attempt to evaluate the adverse effects of other Wi-Fi networks and potentially being able to use 40 MHz channel width, it was decided to perform a recording outside of the building. A greater distance to other networks should reduce their signal strength and influence on the measurements.

All networks detected by the Wi-Fi card were saved with their respective strength and the used channel. They are visualized in Figure 12. Besides the increase in the total number of networks, the average signal strength of other networks on channel six (used for this experiment's measurements) also increased.

This is against expectations and could stem from the missing building structure acting as a dampener for electromagnetic radiation [1]. Neither the 40 MHz channel width nor a potentially less noisy recording environment was achieved.

5.1.3 Automatic Gain Control

While trying to improve the CSI's quality of the recordings, it was discovered that Halperin et al. compensate for Automatic Gain Control (AGC) in their tool. AGC is a general technique used in amplifiers and consists of automatically adjusting the amplification gain so that the resulting signal stays in an optimal range for further processing. It is applied directly in the receiver, even before CSI is calculated. This carries over changes in amplification gain into the CSI and can cause sudden changes in amplitude [34]. It is generally possible to reverse the effects of AGC as long as the gain factor for each frame's CSI is known. Not all hardware and CSI tools make this data available.

The necessary in-depth information required to implement AGC compensation in the Linux CSI Tool was acquired based on Halperin et al.'s employment at Intel during the time of development [14]. The Atheros CSI tool, on the other hand, is reverse-engineered from the usage of the chip with the ath9k driver [48]. Due to the differing levels of knowledge and materials about the Wi-Fi chips and, therefore, features of the CSI tools, it cannot be compensated for AGC in this experiment.

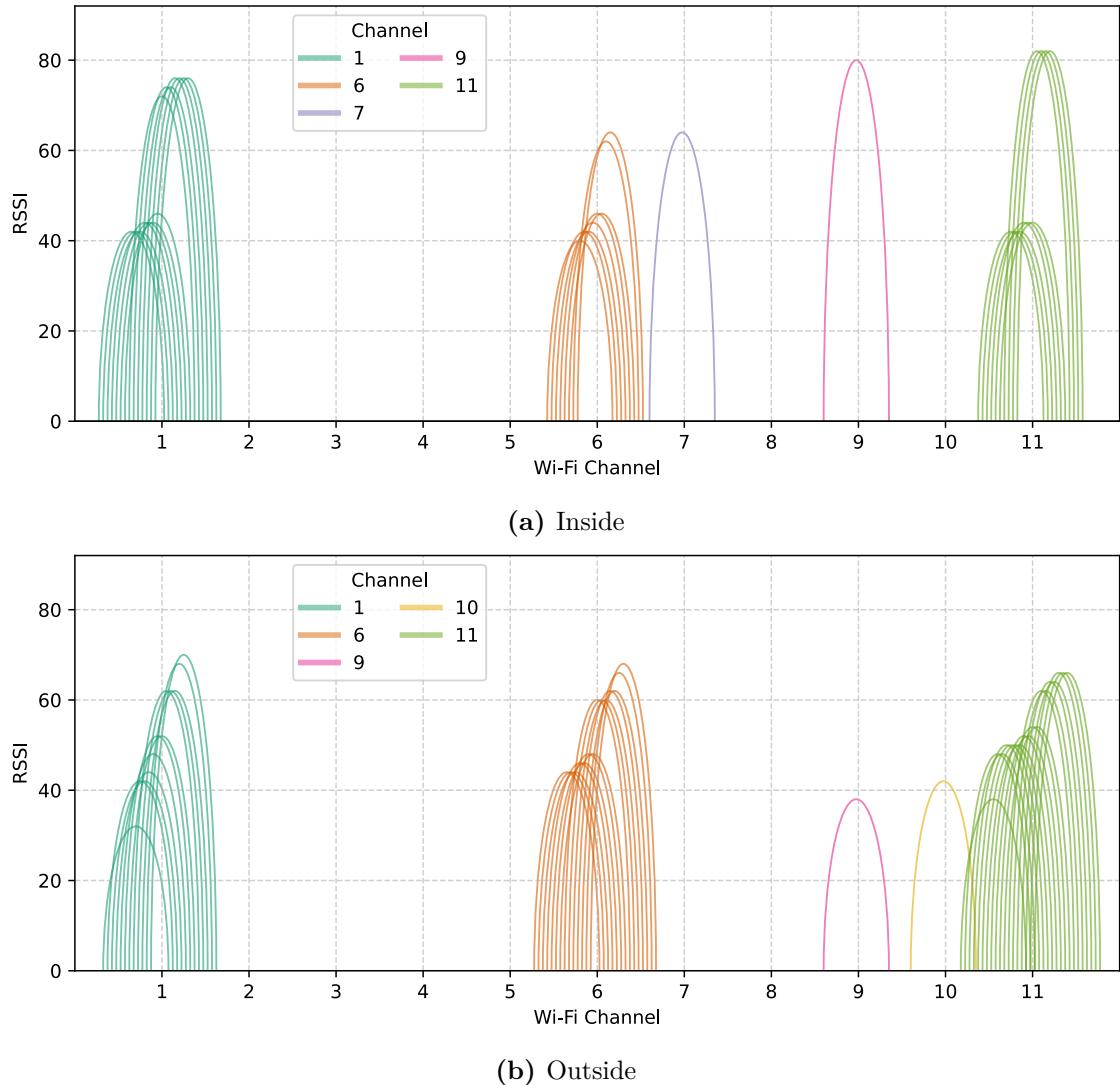


Figure 12. Ambient Wi-Fi Networks Inside and Outside the Building

Table 3. Carrier Groupings in CSI [22]

Grouping Level	Subcarrier Count (20 MHz/40 MHz)	Coverage
1	56/114	all
2	30/58	≈ every 2nd
4	16/30	≈ every 4th

5.2 Modifications to DeepSeg

5.2.1 DeepSeg Improvements

While replicating DeepSeg, several improvements were implemented.

Upgrading Tool Versions The codebase was upgraded to the following versions: Matlab R2018a → R2024a, Python 3.5 → 3.11, and TensorFlow 1.8 → 2.18.

Linux Support Unlike the project’s website suggested, DeepSeg was not initially compatible with Linux and required adjustments.

Code Quality Customizability and adherence to basic programming principles

5.2.2 Data Transformation

The DeepSeg preprocessing pipeline had to be modified in some places to work with the newly collected data.

The different data acquisition methods and resulting shapes required transformation of the collected CSI. Usage of the Atheros CSI Tool resulted in CSI for 3×3 antenna pairs and 56 subcarriers. To fit the given shape of [3 APs \times 30 subcarriers], the amount of data had to be reduced.

The Wi-Fi specification defines different detail levels of CSI reporting [22]. With grouping, only one CSI value is reported for each group of carriers. The number of values is roughly scaled down proportional to the grouping level. The exact numbers are listed in Table 3. In addition to the number of values, the carriers for which information is provided at each grouping level (omitted in the table) are specified. Because the number of values reported by the Linux CSI Tool matches the number of results with grouping level 2 (30), a connection to grouping level 2 was assumed. Therefore, the subcarriers specified by grouping level 2 were selected from the dataset captured with the Atheros CSI Tool.

The higher number of collected APs has to be reduced to the same as in DeepSeg’s segmentation and classification model. As a simple but not optimal solution, the CSI amplitude of each antenna pair is inspected. Three APs with the best visual correspondence to the performed activities are selected.

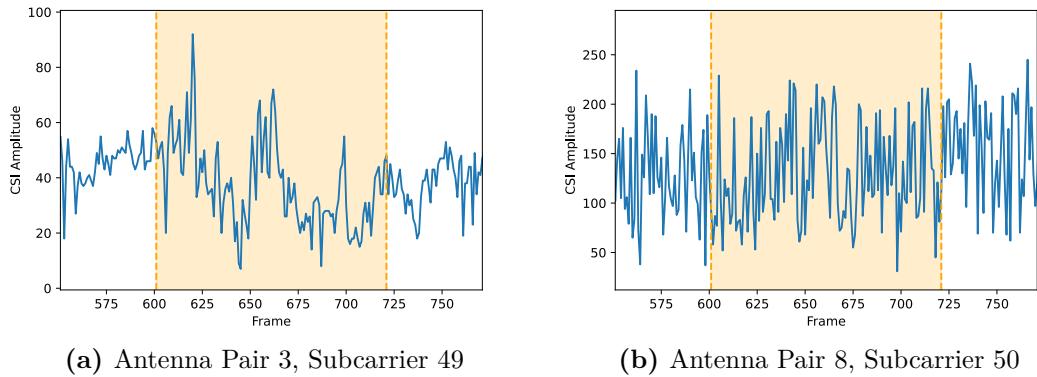


Figure 13. Difference in CSI Amplitude During Activity

As mentioned in 4.4, the new dataset’s recorded activities are shorter than DeepSeg’s. This requires adjustment of the input size and hidden layers of DeepSeg’s CNNs. The stride of convolutional layers and the pool size of the max pooling layer were adjusted. Although not explicitly stated, this was likely also done in DeepSeg when evaluating the effects of different window sizes on model accuracy.

5.3 Extend Collected Data

To compensate for the scarcity of recorded users and attempt to improve the model’s generalization capabilities, an additional data transformation step was created. Instead of only using manually selected antenna pairs (APs) for training, many more samples were generated by including data from all APs. This process involves selecting three APs that, after visual inspection, adequately represent the changes in the CSI that the user’s activities had introduced. The impact of antenna pair and subcarrier choice can be seen in Figure 13. In Figure 13a, the labeled part marking the activity visibly corresponds to changes in CSI. No such substantial change is perceived in Figure 13b.

Picking two of the promising APs and adding a third one out of the complete set of recorded APs gives us 19 different AP combinations. The order in which the selected APs will be combined into a set of three, forming a suitable CSI structure for the model’s training, is randomized. With the manually selected APs as s , the resulting combinations can be represented as

$$\{\text{shuffle}(a, b, c) | a \in s, b \in s, c \in \{1, \dots, 9\}, a \neq b, a \neq c, b \neq c\}$$

CHAPTER 6

Results

The performance of both adjusted models (segmentation and classification) was evaluated on the newly collected dataset. As the results are compared to the baseline models of DeepSeg, whose primary focus was on mixed-granularity activity segmentation, we will only evaluate this type and omit fine- and coarse-only activity constellations. Training hyperparameters such as learning rate, batch size, CNN parameters, etc, were retained from DeepSeg’s codebase unless specified. Testing accuracies for results measured during our training are based on the average accuracy over several epochs after improvements stagnated. All training runs can be found in Table 4 and Table 5 in Appendix A.

6.1 Sensing Performance

For activity segmentation, the modified model initially reached an accuracy of at best 90.1% compared to a claimed $\sim 94\%$ of DeepSeg. Activity recognition was found to be at only 60.4% compared to $\sim 94\%$.

6.1.1 Mixing of Antenna Pairs

The performance changed slightly with the advanced antenna pair selection as detailed in 5.3. Segmentation accuracy decreased to 88.7%, while classification performance increased drastically to 87.7%.

6.2 DeepSeg Performance Robustness

After DeepSeg’s classification results were not reached by a significant amount, we trained the unmodified DeepSeg NNs on the DeepSeg dataset to validate the published numbers by Xiao et al. This resulted in an unexpectedly high segmentation accuracy of 97.7% and low classification accuracy of 76.7%.

Analytical investigation of the cause for this behavior led to experimentation with preprocessing and training parameters. Different random number generator

seeds, training and testing data splits, and training data amounts were tested. Even though DeepSeg's code used a fixed seed in the model training process, there was no significant difference in performance after modifying it.

6.2.1 Test Data Dependence

By default, DeepSeg uses the 6th recorded sequence of every user and activity as the testing data, while sequences one to five are the training set. We changed the test sequence number because this selection seemed unusually specific, and our dataset is missing a recording number six of type "iw". This resulted in extraordinarily different performances.

DeepSeg's performance with a limited number of randomly selected test sequence numbers varied from 33.0% to 97.7% for segmentation and between 62.0% and 76.7% for activity classification.

Applied to our dataset and modified model, this test resulted in accuracies of 46.4% to 90.1% and 22.92% to 60.1% for the simple antenna selection. Advanced antenna selection produced results of segmentation accuracy from 61.6% to 88.7% and classification accuracy from 60.4% to 72.0%.

6.2.2 Reduced DeepSeg Training Data

To ensure an equitable comparison regarding the amount of training data of the modified models against DeepSeg's, the dataset was artificially reduced to only contain the first user. With this reduction, DeepSeg's accuracy dropped to 69.9% in segmentation and 48.6% in classification. An overall effect of the effects on modifying the training data amount can be seen in Figure 14. For DeepSeg, this consists of reducing the dataset from all users to one. For the models using data collected by the Atheros CSI Tool, the antenna pair mixing approach is used. The original dataset is marked as "ownUser1", the generated dataset as "EXT" in the plot.

6.2.3 Hyperparameter optimization

When adjusting the stride parameters of the CNNs for different window sizes, multiple combinations result in compatible tensor shapes. By modifying the parameters of the classification model from $[3 \times 2]$, $[4 \times 2]$ to $[2 \times 2]$, $[5 \times 2]$ along with adjustment of the max-pooling layer's size parameter from $[7 \times 6]$ to $[8 \times 6]$, an accuracy of 87.7% was achieved. This represents the highest classification performance of all conducted measurements across the numerous variations of DeepSeg's and our modified model.

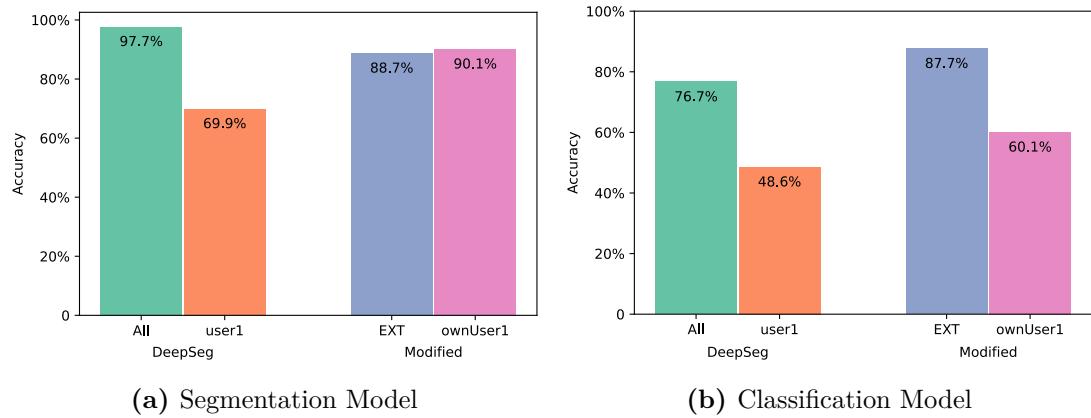


Figure 14. Effects of Different Data Amounts in Training, Highest Achieved Accuracies

CHAPTER 7

Discussion

This chapter discusses the results and findings of training a modified model based on DeepSeg on the collected data, as well as the original model, and explores their connection to the research question.

It was observed that for all variations of training parameters for both neural networks, the training accuracy reached very high values. Regularly, an accuracy of more than 98% or even 100% was reached, effectively halting training. The presence of a very high training accuracy, even though testing accuracies regularly fall short of that by over 20%, indicates overfitting. An example of the progression of training and testing accuracy during training can be seen in Figure 15. The accuracy curves separate early in training, and testing accuracy fluctuates around $\sim 72\%$ before the training accuracy reaches 100% and stops further improvements. Acquiring a more extensive dataset would likely help reduce the accuracy discrepancies, as the CNN has a high number of layers and parameters for comparatively small datasets.

7.1 Training Data Sensitivity

During our review of DeepSeg’s code, we discovered a mismatch between the workings of the code and the reported value for the train/test split ratio. The published ratio by Xiao et al. is 4 : 1. In the code section responsible for creating the training and testing set [12], the training sequences from files ending in `_6.mat` are added to the testing set. Files ending with `_1.mat` to `_5.mat` are combined into the training set. Because the recorded data is equally distributed across the six recordings for each user and sequence type, this results in an effective split ratio of 5 : 1. At the time of model training, only a permutation operation is performed on each individual set, but no redistribution between them [11, 10].

Based on the results in subsection 6.2.1, it is visible that the selection of the train/test split from the dataset is of high importance. The varying results imply considerably non-uniform data, as the impact of varying data selection strategies is higher with data containing more diverse samples. Undocumented changes in training parameters can severely affect the models’ performance and lead to false assumptions about the data selection process. A split of 4 : 1 with utilization of the entire dataset would outrule the current kind of test set creation and suggest a more rigorous approach.

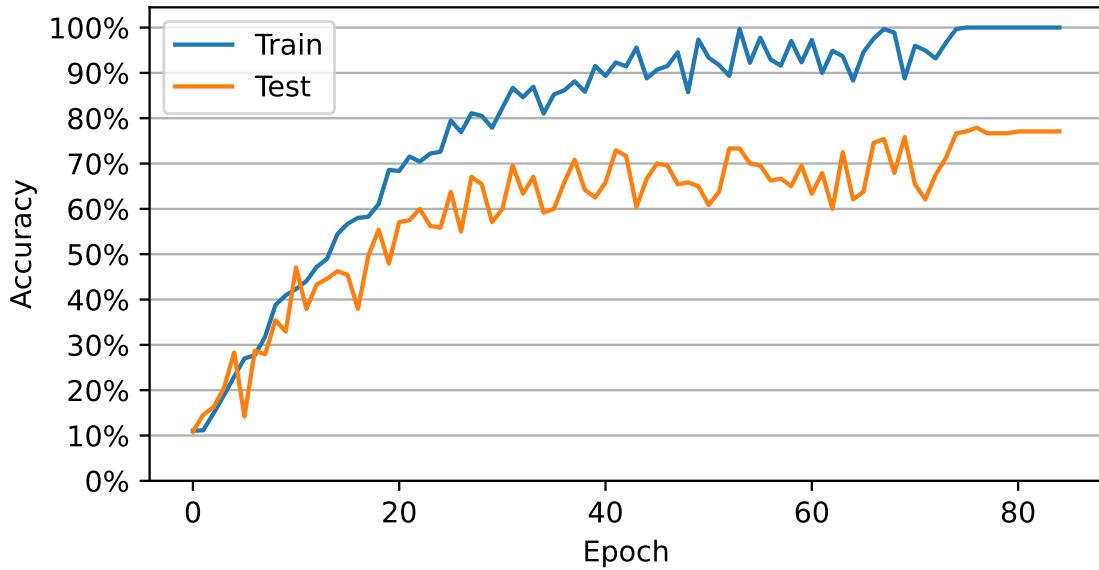


Figure 15. Classification Model Accuracy on Train and Test Set During Training
DeepSeg, All users, Test Sequence 6, Fixed Seed 31288432, Window Size 200

While this can damage the model’s performance, some variation in the dataset is needed to achieve solid generalization capabilities and avoid overfitting to a specific user, setup, or environment. A commonly used method in machine learning to reduce this unwanted behavior is cross-validation [26]. Cross-validation with a partially randomized train/test split instead of a fixed n -th sequence for test data could lead to better robustness of measured performance metrics.

DeepSeg’s performance drops significantly in conditions with reduced training data size. Xiao et al. also experimented with reduced data sizes but did not specify by which method. It is unknown whether their results with 20% training data used samples from all five users or by reducing the number of users from five to one, as with our adjustments. They claimed accuracy of $\sim 88\%$ and $\sim 89\%$ for segmentation and recognition, respectively. Results for reducing data to 20% by only utilizing one user’s recordings were evaluated to be significantly lower at 69.9% and 48.6%.

Unlike our development efforts, Xiao et al. did not base their research and development efforts on a single-user dataset, so significantly lower performance for this task is to be expected.

7.2 Generalisability of Segmentation Methods

This work aimed to explore the generalisability of segmentation methods in Wi-Fi Sensing by testing the compatibility of different hardware with existing segmentation techniques. Our experiments indicate that in the case of DeepSeg, modifications to the data collection process, preprocessing, and physical environment still create

viable results. However, certain limitations were identified that limit the scope of our findings.

7.3 Limitations

Despite the results and accomplishments of this research, several limitations should be acknowledged.

7.3.1 Insufficient User Variety

Due to time constraints, the CSI of activities could only be collected for one user. As indicated by DeepSeg’s improved performance with more user data and the probable model overfitting during training, more data would likely improve performance and help generalize the model for more varying conditions.

7.3.2 Data Quality

Inspection of the data during the labeling process revealed that the activity and idle duration were below the target value, especially in the earlier recordings. This leads to non-uniform activity lengths that are challenging to compensate with a fixed window size. Xiao et al. showed that performance is stable over a window size range of about $\pm 25\%$ of the optimum. However, they showed that by the nature of the segmentation algorithm’s extraction of the four activity sub-states, window sizes approaching the individual activity’s length are problematic. This is because they will include not only the target sub-state but also the CSI of neighboring states. Another natural limit of window sizes is the static-state length. If the static-state is too short, the previous or following activity will be partially included. Window sizes falling short of sufficiently capturing their target state will be more susceptible to noise because of reduced context.

The challenging activity length conditions in the created dataset assumably harm model performance. This could be mitigated in future recordings.

7.3.3 CSI Antenna and Subcarrier Selection

For simplicity, the initial choice of the antenna pairs selected for use was made only by visual inspection. The advanced method, which includes mixing of further antenna pairs, still relies on a manual preselection. This approach lacks clear decision criteria and requires previous knowledge. Finding a different approach is desirable and could potentially improve performance through a larger volume of data with

more variety. It should be noted that as Chen et al. observed with LightSeg [9], including data of lower quality antenna pairs can also reduce model performance.

Subcarrier selection was closely oriented at the CSI carrier grouping mechanism. Further improvements could be achieved by not using equal spacing across the channel frequency range but instead focusing on subcarriers carrying the most usable information. A suitable metric for subcarrier grading is needed.

7.3.4 Joint Training of Segmentation and Classification

Joint training for both models could not be conducted due to existing errors and the complexity of DeepSeg’s codebase, which posed development challenges. Although it did not yield exceptional improvements, it benefited DeepSeg and displays possibilities for further exploration.

7.3.5 Wi-Fi Interference and Frame Timings

The occurrence of non-constant frame rates, as observed in the experiments, causes uneven coverage of the activity. Gaps in the recording might miss crucial information about the activity. Other Wi-Fi networks in the surroundings likely had a negative effect of unknown magnitude on our data despite creating a more realistic recording environment. They also prevented using the 40 MHz channel width and the jointly increased subcarrier count. As research in the field usually does not provide sufficiently detailed information about the Wi-Fi environment, judging the conditions of this experiment and other works is challenging.

7.4 Future Work

The concept of cross-platform use of CSI for segmentation was successfully demonstrated in this experiment. However, further efforts are needed to address the limitations and expand its applicability.

Most importantly, a larger dataset with more users should be collected to enable a more meaningful and accurate comparison to DeepSeg. Additionally, the increased amount of CSI gained through the selection of a more capable Wi-Fi platform was not optimally utilized. Future research could explore advanced preprocessing, data transformations, or alternative model architectures to enhance segmentation performance in this application further.

7.5 Conclusion

This work serves as a proof of concept for enhancing the generalization of segmentation methods in Wi-Fi Sensing by demonstrating the interchangeable use of hardware on a purpose-built model. We collected activity data in a different environment by means of different hardware and CSI Tooling. The necessary data transformation steps to use the foreign data with the existing preprocessing and training pipeline were carried out. This new approach was evaluated and demonstrated to have comparable performance to the original model on its respective training data.

A strong dependence of the model architecture on the train/test data selection uniformity at these limited dataset sizes was identified. Performance is unstable in regard to small changes in the training parameters and activity execution. This highlights the need for more cross-environment segmentation datasets to enable sufficient generalization of the neural network-based approach and prevent overfitting to specific conditions.

A method of leveraging the platform-specific higher amount of collected data was developed, introducing additional variety into the training data and leveraging the larger amount of data collected by the chosen Wi-Fi Sensing system.

Overall, our experiment showed the viability of segmentation in Wi-Fi Sensing through the use of cross-platform data collection. This is a valuable step towards more generalized segmentation techniques, which would advance Wi-Fi Sensing as a whole through increased real-world usability. With better segmentation capabilities, sensing can be carried out continuously and with higher efficiency and performance, thus enabling its use outside of lab environments.

Future research could focus on the creation of more advanced datasets containing a larger variety of environments and overall data. As models would be able to learn the general concepts needed for segmentation instead of focusing on environment and setup-specific features, better universal sensing could be achieved.

Bibliography

- [1] Fadel Adib and Dina Katabi. “See through walls with WiFi!” In: *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*. 2013, pp. 75–86.
- [2] Kamran Ali et al. “Keystroke Recognition Using WiFi Signals”. In: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. MobiCom’15: The 21th Annual International Conference on Mobile Computing and Networking. Paris France: ACM, Sept. 7, 2015, pp. 90–102. DOI: [10.1145/2789168.2790109](https://doi.org/10.1145/2789168.2790109).
- [3] Kamran Ali et al. “Recognizing Keystrokes Using WiFi Devices”. In: *IEEE Journal on Selected Areas in Communications* 35.5 (May 2017), pp. 1175–1190. DOI: [10.1109/JSAC.2017.2680998](https://doi.org/10.1109/JSAC.2017.2680998).
- [4] Samaneh Aminikhanghahi and Diane J. Cook. “A Survey of Methods for Time Series Change Point Detection”. In: *Knowledge and information systems* 51.2 (May 2017), pp. 339–367. DOI: [10.1007/s10115-016-0987-z](https://doi.org/10.1007/s10115-016-0987-z).
- [5] Samaneh Aminikhanghahi and Diane J. Cook. “Enhancing activity recognition using CPD-based activity segmentation”. In: *Pervasive and Mobile Computing* 53 (Feb. 2019), pp. 75–89. DOI: [10.1016/j.pmcj.2019.01.004](https://doi.org/10.1016/j.pmcj.2019.01.004).
- [6] Samaneh Aminikhanghahi, Tinghui Wang, and Diane J. Cook. “Real-Time Change Point Detection with Application to Smart Home Time Series Data”. In: *IEEE Transactions on Knowledge and Data Engineering* 31.5 (May 2019), pp. 1010–1023. DOI: [10.1109/TKDE.2018.2850347](https://doi.org/10.1109/TKDE.2018.2850347).
- [7] Qirong Bu et al. “Deep transfer learning for gesture recognition with WiFi signals”. In: *Personal and Ubiquitous Computing* (2022), pp. 1–12.
- [8] S. Chakar et al. “A robust approach for estimating change-points in the mean of an AR(1) process”. In: *Bernoulli* 23.2 (May 2017), pp. 1408–1447. DOI: [10.3150/15-BEJ782](https://doi.org/10.3150/15-BEJ782).
- [9] Liming Chen et al. “LightSeg: An Online and Low-Latency Activity Segmentation Method for Wi-Fi Sensing”. In: *Mobile and Ubiquitous Systems: Computing, Networking and Services*. Ed. by Shangguan Longfei and Priyantha Bodhi. Cham: Springer Nature Switzerland, 2023, pp. 227–247. DOI: [10.1007/978-3-031-34776-4_13](https://doi.org/10.1007/978-3-031-34776-4_13).

- [10] Xiao Chunjing et al. *DeepSeg Train/Test Set Loading Classification*. URL: https://github.com/ChunjingXiao/DeepSeg/blob/a0f0d115b427f328280c4df4ae7ed2b6b06Classification_ClassifyActivity/1train_4_200_30_3_Mat2npy_Weight.py#L154.
- [11] Xiao Chunjing et al. *DeepSeg Train/Test Set Loading Segmentation*. URL: https://github.com/ChunjingXiao/DeepSeg/blob/a0f0d115b427f328280c4df4ae7ed2b6b04Segment_StateInference/1train_4_200_30_3_Mat2npy_Weight_Segm.py#L154.
- [12] Xiao Chunjing et al. *DeepSeg Train/Test Set Sample Selection*. URL: https://github.com/ChunjingXiao/DeepSeg/blob/a0f0d115b427f328280c4df4ae7ed2b6be1bac3502Segment_ExtractTrainData/a2_segmentCombineTrainData.m#L64.
- [13] Neena Damodaran et al. “Device free human activity and fall recognition using WiFi channel state information (CSI)”. In: *CCF Transactions on Pervasive Computing and Interaction* 2.1 (Mar. 1, 2020), pp. 1–17. DOI: [10.1007/s42486-020-00027-1](https://doi.org/10.1007/s42486-020-00027-1).
- [14] dhalperi. *Linux 802.11n CSI Tool: Employment at Intel*.
- [15] Jianyang Ding and Yong Wang. “WiFi CSI-Based Human Activity Recognition Using Deep Recurrent Neural Network”. In: *IEEE Access* 7 (2019), pp. 174257–174269. DOI: [10.1109/ACCESS.2019.2956952](https://doi.org/10.1109/ACCESS.2019.2956952).
- [16] Felix Dobler. *Evaluating the Generalisability of Segmentation Methods in WiFi-Sensing*. Dec. 13, 2024. URL: <https://github.com/FelixDobler/BachelorThesis-WiFi-Sensing-Segmentation>.
- [17] Parisa Fard Moshiri et al. “A CSI-Based Human Activity Recognition Using Deep Learning”. In: *Sensors* 21.21 (Jan. 2021), p. 7225. DOI: [10.3390/s21217225](https://doi.org/10.3390/s21217225).
- [18] Chunhai Feng et al. “Wi-Multi: A Three-Phase System for Multiple Human Activity Recognition With Commercial WiFi Devices”. In: *IEEE Internet of Things Journal* 6.4 (Aug. 2019), pp. 7293–7304. DOI: [10.1109/JIOT.2019.2915989](https://doi.org/10.1109/JIOT.2019.2915989).
- [19] Yao Ge et al. “Contactless WiFi Sensing and Monitoring for Future Healthcare - Emerging Trends, Challenges, and Opportunities”. In: *IEEE Reviews in Biomedical Engineering* 16 (2023), pp. 171–191. DOI: [10.1109/RBME.2022.3156810](https://doi.org/10.1109/RBME.2022.3156810).
- [20] Daniel Halperin et al. “Tool release: gathering 802.11n traces with channel state information”. In: *ACM SIGCOMM Computer Communication Review* 41.1 (Jan. 22, 2011), p. 53. DOI: [10.1145/1925861.1925870](https://doi.org/10.1145/1925861.1925870).
- [21] Zhanjun Hao et al. “Wi-SL: Contactless Fine-Grained Gesture Recognition Uses Channel State Information”. In: *Sensors* 20.14 (Jan. 2020), p. 4025. DOI: [10.3390/s20144025](https://doi.org/10.3390/s20144025).
- [22] IEEE. *IEEE Standard 802.11n-2009*. 2009.

- [23] Md Shafiqul Islam et al. “STC-NLSTMNet: An Improved Human Activity Recognition Method Using Convolutional Neural Network with NLSTM from WiFi CSI”. In: *Sensors* 23.1 (Jan. 2023), p. 356. DOI: 10.3390/s23010356.
- [24] Mir Kanon Ara Jannat et al. “Efficient Wi-Fi-Based Human Activity Recognition Using Adaptive Antenna Elimination”. In: *IEEE Access* 11 (2023), pp. 105440–105454. DOI: 10.1109/ACCESS.2023.3320069.
- [25] Jianguo Jiang et al. “Wi-Gait: Pushing the limits of robust passive personnel identification using Wi-Fi signals”. In: *Computer Networks* 229 (June 2023), p. 109751. DOI: 10.1016/j.comnet.2023.109751.
- [26] R Kohavi. “A study of cross-validation and bootstrap for accuracy estimation and model selection”. In: *Morgan Kaufman Publishing* (1995).
- [27] Fan Li et al. “ClickLeak: Keystroke Leaks Through Multimodal Sensors in Cyber-Physical Social Networks”. In: *IEEE Access* 5 (2017), pp. 27311–27321. DOI: 10.1109/ACCESS.2017.2776527.
- [28] Jiguang Lv et al. “Robust WLAN-Based Indoor Intrusion Detection Using PHY Layer Information”. In: *IEEE Access* 6 (2018), pp. 30117–30127. DOI: 10.1109/ACCESS.2017.2785444.
- [29] Yongsen Ma, Gang Zhou, and Shuangquan Wang. “WiFi Sensing with Channel State Information: A Survey”. In: *ACM Computing Surveys* 52.3 (June 18, 2019), 46:1–46:36. DOI: 10.1145/3310194.
- [30] Yongsen Ma, Gang Zhou, and Shuangquan Wang. “WiFi Sensing with Channel State Information: A Survey”. In: *ACM Computing Surveys* 52.3 (May 31, 2020), pp. 1–36. DOI: 10.1145/3310194.
- [31] Yongsen Ma et al. “SignFi: Sign Language Recognition Using WiFi”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2.1 (Mar. 26, 2018), pp. 1–21. DOI: 10.1145/3191755.
- [32] Francesca Meneghelli et al. “SHARP: Environment and Person Independent Activity Recognition With Commodity IEEE 802.11 Access Points”. In: *IEEE Transactions on Mobile Computing* 22.10 (Oct. 2023), pp. 6160–6175. DOI: 10.1109/TMC.2022.3185681.
- [33] Li Mo and Xie Yaxiong. *Atheros CSI Tool*. 2015. URL: <https://wands.sg/research/wifi/AtherosCSI/>.
- [34] Jörg Schäfer et al. “Human Activity Recognition Using CSI Information with Nexmon”. In: *Applied Sciences* 11.19 (Sept. 23, 2021), p. 8860. DOI: 10.3390/app11198860.
- [35] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. *Nexmon: The C-based Firmware Patching Framework*. 2017. URL: <https://nexmon.org>.

- [36] Eman Shalaby, Nada ElShennawy, and Amany Sarhan. “Utilizing deep learning models in CSI-based human activity recognition”. In: *Neural Computing and Applications* 34.8 (Apr. 1, 2022), pp. 5993–6010. DOI: 10.1007/s00521-021-06787-w.
- [37] Jiacheng Shang and Jie Wu. “A Robust Sign Language Recognition System with Multiple Wi-Fi Devices”. In: *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture*. SIGCOMM ’17: ACM SIGCOMM 2017 Conference. Los Angeles CA USA: ACM, Aug. 11, 2017, pp. 19–24. DOI: 10.1145/3097620.3097624.
- [38] Biyun Sheng et al. “Deep spatial-temporal model based cross-scene action recognition using commodity WiFi”. In: *IEEE Internet of Things Journal* 7.4 (2020), pp. 3592–3601.
- [39] Zhenguo Shi et al. “Environment-Robust Device-Free Human Activity Recognition With Channel-State-Information Enhancement and One-Shot Learning”. In: *IEEE Transactions on Mobile Computing* 21.2 (2020), pp. 540–554. DOI: 10.1109/TMC.2020.3012433.
- [40] Aditya Virmani and Muhammad Shahzad. “Position and Orientation Agnostic Gesture Recognition Using WiFi”. In: *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys’17: The 15th Annual International Conference on Mobile Systems, Applications, and Services. Niagara Falls New York USA: ACM, June 16, 2017, pp. 252–264. DOI: 10.1145/3081333.3081340.
- [41] Wei Wang, Alex X. Liu, and Muhammad Shahzad. “Gait recognition using wifi signals”. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. UbiComp ’16: The 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing. Heidelberg Germany: ACM, Sept. 12, 2016, pp. 363–373. DOI: 10.1145/2971648.2971670.
- [42] Xuyu Wang, Chao Yang, and Shiwen Mao. “PhaseBeat: Exploiting CSI Phase Data for Vital Sign Monitoring with Commodity WiFi Devices”. In: *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*. 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). Atlanta, GA, USA: IEEE, June 2017, pp. 1230–1239. DOI: 10.1109/ICDCS.2017.206.
- [43] Zhengjie Wang et al. “A Survey on Human Behavior Recognition Using Channel State Information”. In: *IEEE Access* 7 (2019), pp. 155986–156024. DOI: 10.1109/ACCESS.2019.2949123.
- [44] Xuangou Wu et al. “TW-See: Human activity recognition through the wall with commodity Wi-Fi devices”. In: *IEEE Transactions on Vehicular Technology* 68.1 (2018), pp. 306–319.

- [45] Chunjing Xiao et al. “CsiGAN: Robust Channel State Information-Based Activity Recognition With GANs”. In: *IEEE Internet of Things Journal* 6.6 (Dec. 2019), pp. 10191–10204. doi: [10.1109/JIOT.2019.2936580](https://doi.org/10.1109/JIOT.2019.2936580).
- [46] Chunjing Xiao et al. “DeepSeg: Deep-Learning-Based Activity Segmentation Framework for Activity Recognition Using WiFi”. In: *IEEE Internet of Things Journal* 8.7 (Apr. 1, 2021), pp. 5669–5681. doi: [10.1109/JIOT.2020.3033173](https://doi.org/10.1109/JIOT.2020.3033173).
- [47] Yaxiong Xie, Zhenjiang Li, and Mo Li. “Precise Power Delay Profiling with Commodity WiFi”. In: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. MobiCom’15: The 21th Annual International Conference on Mobile Computing and Networking. Paris France: ACM, Sept. 7, 2015, pp. 53–64. doi: [10.1145/2789168.2790124](https://doi.org/10.1145/2789168.2790124).
- [48] xieyaxiongfly. *Uncertainty of CSI origin in Atheros CSI Tool*.
- [49] Santosh Kumar Yadav et al. “CSITime: Privacy-preserving human activity recognition using WiFi channel state information”. In: *Neural Networks* 146 (Feb. 2022), pp. 11–21. doi: [10.1016/j.neunet.2021.11.011](https://doi.org/10.1016/j.neunet.2021.11.011).
- [50] Zheng Yang. *Widar3.0 Dataset: Cross-Domain Gesture Recognition with WiFi*. Dec. 26, 2020. doi: [10.21227/7ZNF-QP86](https://doi.org/10.21227/7ZNF-QP86).
- [51] Zheng Yang, Zimu Zhou, and Yunhao Liu. “From RSSI to CSI: Indoor localization via channel response”. In: *ACM Computing Surveys* 46.2 (Nov. 2013), pp. 1–32. doi: [10.1145/2543581.2543592](https://doi.org/10.1145/2543581.2543592).
- [52] Andrii Zhuravchak, Oleg Kapshii, and Evangelos Pournaras. “Human Activity Recognition based on Wi-Fi CSI Data -A Deep Neural Network Approach”. In: *Procedia Computer Science* 198 (2021), pp. 59–66. doi: [10.1016/j.procs.2021.12.211](https://doi.org/10.1016/j.procs.2021.12.211).

CHAPTER A

Training Parameters and Performance

Table 4. Segmentation Model

Model	Antenna Selection	Users	Test Sequence	Seed	Window Size	Stride1	Stride2	Pool Size	Accuracy
DeepSeg	-	All	6	given (10)	120	4,2	4,2	6,6	97.7%
DeepSeg	-	All	6	31288432	120	4,2	4,2	6,6	97.7%
DeepSeg	-	All	5	given (10)	120	4,2	4,2	6,6	33.0%
DeepSeg	-	All	3	given (10)	120	4,2	4,2	6,6	34.9%
DeepSeg	-	user1	6	31288432	120	4,2	4,2	6,6	56.4%
DeepSeg	-	user1	3	31288432	120	4,2	4,2	6,6	69.9%
Modified	Simple	ownUser1	6	given (10)	60	2,2	4,2	6,6	90.1%
Modified	Simple	ownUser1	5	given (10)	60	2,2	4,2	6,6	88.8%
Modified	Simple	ownUser1	3	given (10)	60	2,2	4,2	6,6	46.4%
Modified	Advanced	EXT	6	given (10)	60	2,2	4,2	6,6	88.7%
Modified	Advanced	EXT	5	given (10)	60	2,2	4,2	6,6	85.8%
Modified	Advanced	EXT	3	given (10)	60	2,2	4,2	6,6	61.6%

Table 5. Classification Model

Model	Antenna Selection	Users	Test Sequence	Seed	Window Size	Stride1	Stride2	Pool Size	Accuracy
DeepSeg	-	All	6	given (10)	200	5,2	5,2	6,6	76.7%
DeepSeg	-	All	6	31288432	200	5,2	5,2	6,6	72.2%
DeepSeg	-	All	5	given (10)	200	5,2	5,2	6,6	75.6%
DeepSeg	-	All	3	31288432	200	5,2	5,2	6,6	59.5%
DeepSeg	-	All	3	given (10)	200	5,2	5,2	6,6	62.0%
DeepSeg	-	user1	6	31288432	200	5,2	5,2	6,6	48.6%
DeepSeg	-	user1	3	31288432	200	5,2	5,2	6,6	48.0%
Modified	Simple	ownUser1	5	31288432	100	3,2	4,2	7,6	60.1%
Modified	Simple	ownUser1	3	31288432	100	3,2	4,2	7,6	22.92%
Modified	Advanced	EXT	5	given (10)	100	3,2	4,2	7,6	60.4%
Modified	Advanced	EXT	3	given (10)	100	3,2	4,2	7,6	72.0%
Modified	Advanced	EXT	3	given (10)	100	2,2	5,2	8,6	87.7%